Класове на сложност **P** и **NP**

Тодор Дуков

Какво имаме предвид под класове на сложност?

При решаването на различни видове задачи, които ни интересуват, е естествено да се опитаме да ги класифицираме по това колко са *"сложни"*. Това сме го виждали вече – в курса по ЕАИ сме класифицирали различни езици спрямо това каква машина може да решава въпроса за принадлежност към съответния език. Тук ще направим нещо подобно, разликата ще бъде в това, че ще се интересуваме от това за какво време се решава една задача.

- Класът на сложност **P** ще бъде множеството от всички изчислителни задачи за разпознаване, за които съществува алгоритъм с полиномиална времева сложност при най-лоши входни данни.
- Класът на сложност **NP*** ще бъде множеството от всички изчислителни задачи за разпознаване, за които съществува алгоритъм с полиномиална времева сложност при всякакви входни данни, проверяващ отговора ДА на задачата с помощта на допълнителен параметър, наречен **сертификат**, който зависи от входните данни на задачата и чиято дължина е полиномиална спрямо тяхната

Нека дадем пример за сертификат. Да кажем, че търсим отговор на въпроса

Вярно ли е, че уравнението $a_0 + a_1x + \dots + a_nx^n = 0$ *има реален корен?*

По-лесно ще бъде да верифицираме някое предложено решение т.е. да отговорим на въпроса

Вярно ли е, че реалното число x_0 *е корен на уравнението* $a_0 + a_1x + \dots + a_nx^n = 0$*?*

В него имаме още един параметър – предложеното решение x_0. В този случай това ще бъде сертификатът. Въпреки, че в примера е направено така, не е нужно сертификата да се използва. Той е само помощен, и съществуването му е нужно само при отговор ДА. Ясно е, че при отговор НЕ няма и как да има сертификат.
Неформално казано, в класа **P** се намират задачите, които се решават *"бързо"* (за полиномиално време), а в класа **NP** се намират задачите, чиито решения се верифицират *"бързо"*. Лесно може да се види, че $\mathbf{P} \subseteq \mathbf{NP}$. Ако една задача може да се реши за полиномиално време без да използва сертификат, то тогава тя може да се реши и със използване на сертификат.

Няколко важни задачи за класа NP

Следните задачи за изключително важни за класа **NP** (по-късно ще разберем защо):

- Задачата **SAT**:

  Вход: Съждителна формула φ в конюнктивна нормална форма.

  Въпрос: Има ли оценка, в която φ е вярна?

- Задачата **3SAT**:

  Вход: Съждителна формула φ в конюнктивна нормална форма, при която във всяка дизюнктивна клауза участват точно три литерала.

  Въпрос: Има ли оценка, в която φ е вярна?

- Задачата **SubsetSum**:

  Вход: Масив $A[1\dots n]$ от положителни числа и естествено число s.

  Въпрос: Има ли $I \subseteq \{1,\dots,n\}$, за което $\sum\limits_{i\in I} A[i] = s$?

- Задачата **2Partition**:

  Вход: Масив $A[1\dots n]$ от положителни числа.

  Въпрос: Има ли $I \subseteq \{1,\dots,n\}$, за което $\sum\limits_{i\in I} A[i] = \sum\limits_{i\in\{1,\dots,n\}\setminus I} A[i]$?

*За този клас има и алтернативна дефиниция – в нея не е нужен сертификат, но се допуска алгоритъмът да е недетерминиран. Идеята е, че той едновременно *"познава"* правилния отговор, и го верифицира.

- Задачата **VertexCover**:

  Вход: Граф $G = \langle V, E \rangle$ и естествено число k.

  Въпрос: Има ли $X \subseteq V$, за което $|X| \leq k$ и X съдържа поне един край на всяко ребро от E?

- Задачата **DominatingSet**:

  Вход: Граф $G = \langle V, E \rangle$ и естествено число k.

  Въпрос: Има ли $X \subseteq V$, за което $|X| \leq k$ и всеки връх от $V \setminus X$ е съседен на някой от X?

- Задачата **Clique**:

  Вход: Граф $G = \langle V, E \rangle$ и естествено число k.

  Въпрос: Има ли клика $X \subseteq V$, за която $|X| \geq k$?

- Задачата **Anticlique**:

  Вход: Граф $G = \langle V, E \rangle$ и естествено число k.

  Въпрос: Има ли антиклика $X \subseteq V$, за която $|X| \geq k$?

- Задачата **HamiltonianPath**:

  Вход: Граф $G = \langle V, E \rangle$ и два върха $s, e \in V$.

  Въпрос: Има ли хамилтонов път в G от s до e?

- Задачата **EulerianPath**:

  Вход: Граф $G = \langle V, E \rangle$ и два върха $s, t \in V$.

  Въпрос: Има ли ойлеров път в G от s до e?

- Задачата **TSP**:

  Вход: Тегловен граф $G = \langle V, E, w \rangle$ и естествено число k.

  Въпрос: Има ли хамилтонов път в G с тегло ненадвишаващо k?

Нека покажем за някои от тях, че попадат в класа **NP**.
Да започнем със **SAT**. Трябва да предложим полиномиален алгоритъм, който с помощта на сертификат проверява за отговор ДА. Сертифкатът ще бъде оценката. Оценката може да се представи като масив от променливи $V[1 \dots k]$, чиито членове са променливите, които имат стойност истина. Ясно е, че тази кодировка е с полиномиална относно формулата дължина. При дадена оценка, лесно можем да видим дали формулата е вярна:

```
SAT(φ - съждителна формула в КНФ, V[1...k] - сертификат):
    за всеки дизюнкт D във φ:
        ако има литерал L в D, за който (L = x и x ∈ V[1...k]) или (L = x̄ и x ∉ V[1...k]):
            продължи нататък
        иначе:
            върни False

    върни True
```

Този алгоритъм очевидно работи за полиномиално време. Наистина, той проверява дали $V[1 \dots k]$ задава оценка, в която φ е вярна. Така **SAT** е в класа **NP**. Тъй като **3SAT** е просто по-лесна версия на **SAT**, тя също попада в класа **NP**.
Нека сега видим, че **SubsetSum** е в класа **NP**. Тук сертификатът ще бъде масив $I[1 \dots k]$, който ще представя множество от индекси за входния масив. При дадено множество от индекси, лесно можем да проверим дали е изпълнено условието за сумата:

```
SubsetSum(A[1...n] - масив от положителни числа, s - естествено число, I[1...k] - сертификат):
    инициализирай S със 0

    за всяко i от 1 до k:
        прибави A[I[i]] към S

    върни дали S = s?
```

Този алгоритъм очевидно работи за полиномиално време. Наистина, той проверява дали $I[1 \dots k]$ задава подмножество I на $\{1, \dots, n\}$, за което $\sum\limits_{i \in I} A[i] = s$. Така **SubsetSum** е в класа **NP**.

Да видим сега, че **VertexCover** е в класа **NP**. Тук сертификатът ще бъде масив $X[1 \dots n]$, който ще представя множеството от върховете, които ще образуват потенциално върхово покритие. При дадено множество от върхове, лесно можем да видим дали то е върхово покритие с размер най-много k:

```
VertexCover(G = (V, E) - граф, k - естествено число, X[1...n] - сертификат):
    ако n > k:
        върни False

    за всяко ребро (u, v) ∈ E:
        ако u ∉ X[1...n] и v ∉ X[1...n]:
            върни False

    върни True
```

Този алгоритъм очевидно работи за полиномиално време. Наистина, той проверява дали $X[1 \dots n]$ задава върхово покритие на G с най-много k елемента. Така **VertexCover** е в класа **NP**.
Сега ще покажем, че **Clique** е в класа **NP**. Тук сертификатът ще бъде масив $X[1 \dots n]$, който ще представя множеството от върховете, които ще образуват потенциална клика. При дадено множество от върхове, лесно можем да видим дали то е клика с размер поне k:

```
Clique(G = (V, E) - граф, k - естествено число, X[1...n] - сертификат):
    ако n < k:
        върни False

    за всеки връх u ∈ X[1...n]:
        за всеки връх v ∈ X[1...n]:
            ако u ≠ v и (u, v) ∉ E:
                върни False

    върни True
```

Този алгоритъм очевидно работи за полиномиално време. Наистина, той проверява дали $X[1 \dots n]$ задава клика в G с поне k елемента. Така **Clique** е в класа **NP**.

Задачи

Задача 1. Да се докаже, че следните задачи са в класа **NP**:

- **2Partition**;
- **DominatingSet**;
- **Anticlique**;
- **HamiltonianPath**;
- **EulerianPath**;
- **TSP**.

Задача 2. Разглеждаме задачата **StarFreeRegexIneq**:

Вход: Два регулярни израза r_1 и r_2, в които не участва $*$.
Въпрос: Вярно ли е, че $\mathcal{L}[\![r_1]\!] \neq \mathcal{L}[\![r_2]\!]$?

Да се докаже, че задачата **StarFreeRegexIneq** е в класа **NP**.

Задача 3. Разглеждаме задачата **ChromaticNumber**:

Вход: Граф $G = \langle V,\, E \rangle$ и естествено число k.
Въпрос: Вярно ли е, че хроматичното число на G не надвишава k?

Да се докаже, че задачата **ChromaticNumber** е в класа **NP**.